**ОТЧЕТ**

по лабораторной работе №5

по курсу «Web-программирование»

на тему «Реализация ответа обработчиков и POST-запросы»

Выполнили:

студенты группы 16ВВ2

Гогулевский А.Э.

Кузнецов А.С.

Мультяева И.И.

Приняли:

к.т.н., доцент Дубравин А.В.

к.т.н., доцент Карамышева Н.С.

Пенза 2020

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Название

Реализация ответа обработчиков и POST-запросы.

### Цель работы

Реализовать ответ обработчиков и POST-запросы.

### Лабораторное задание

a. На стороне клиента реализовать проверку: если в текстовом поле указано одно из трех животных в следующем формате: «antilopa»или «antilopa7» (число от 0 до 9), выполнить формирование и отправку на сервер POST-запроса, в противном случае вывести сообщение о некорректно указанных данных.

b. На стороне сервера реализовать три обработчика, соответствующих каждому из указанных в варианте животных.

c. В обработчике, в случае, если в URLзапросе отсутствует параметр с указанием номера запрашиваемого изображения животного, выполнить генерацию случайного числа в диапазоне от 0 до 9 и выполнить отправку клиенту соответствующего изображения. Если в URLзапросе присутствует параметр с указанием номера запрашиваемого изображения животного – выполнить отправку клиенту соответствующего изображения.

|  |  |
| --- | --- |
| Вариант | Задание |
| 7 | Животные: журавль, жаба, жираф |

### Порядок выполнения работы

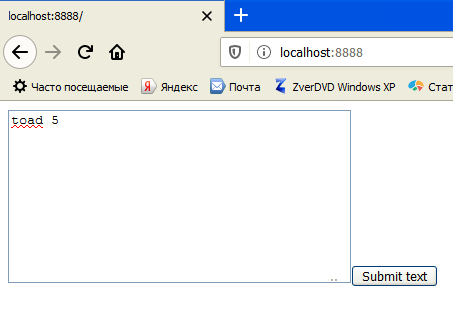


Рисунок 1–ввод сообщения

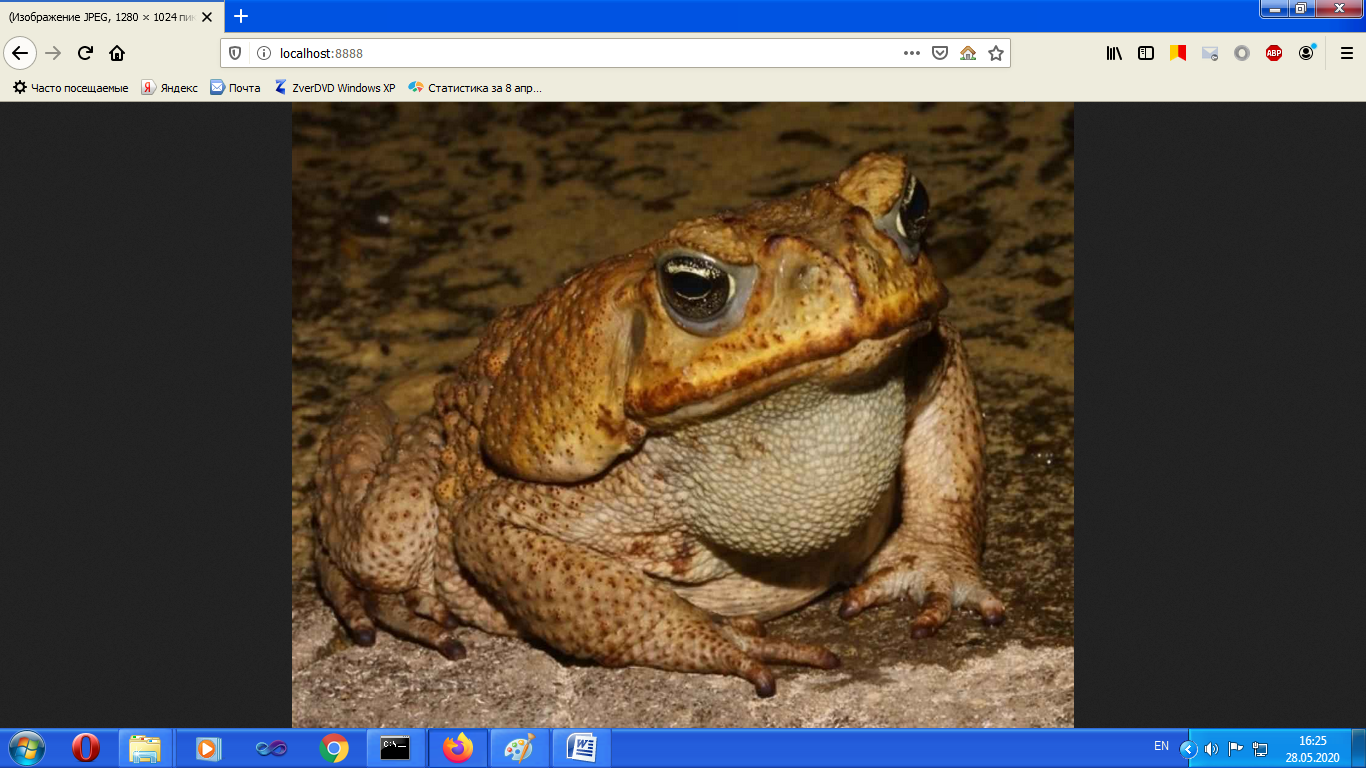


Рисунок 2 - вывод изображения жабы

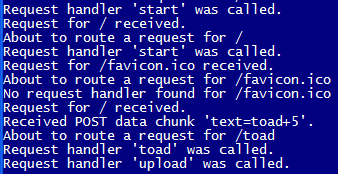


Рисунок 3 - вывод в консоль

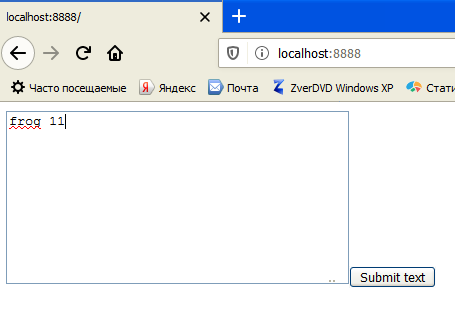


Рисунок 4 – ввод неверных данных

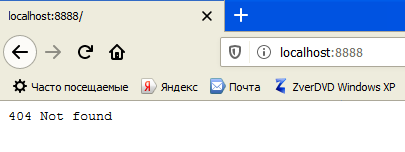


Рисунок 5 – Сообщение об ошибке

### Листинг

#### requestHandler.js

var fs = require("fs");

var query = require('querystring');

function start(response, postData) {

console.log("Request handler 'start' was called.");

var body = '<html>'+

'<head>'+

'<meta http-equiv="Content-Type" content="text/html; '+

'charset=UTF-8" />'+

'</head>'+

'<body>'+

'<form onsubmit="return check()" method="post">'+

'<textarea name="text" rows="10" cols="40"></textarea>'+

'<input type="submit" value="Submit text" />'+

'</form>'+

'</body>'+

'<script>'+

'check();'+

'</script>'+

'</html>';

response.writeHead(200, {"Content-Type": "text/html"});

response.write(body);

response.end();

}

function crane(response, postData) {

console.log("Request handler 'crane' was called.");

upload(response, postData);

}

function toad(response, postData) {

console.log("Request handler 'toad' was called.");

upload(response, postData);

}

function giraffe(response, postData) {

console.log("Request handler 'giraffe' was called.");

upload(response, postData);

}

function check() {

var inputwords = ['crane','toad','giraffe'];

var input = document.getElementById("input");

var inputword = input.value;

var array = inputword.split(' ');

for (var i = 0; i < inputwords.length; i++) {

if (inputword == inputwords[i]) {return true;}

else {

for (var j = 0; j < 10; j++) {

if (inputword == inputwords[i] + ' ' + j) {return true;}

}

}

}

alert ("Неверный ввод данных");

return false;

}

function upload(response, postData) {

console.log("Request handler 'upload' was called.");

var data = query.parse(postData).text;

var array = data.split(' ');

var img;

response.writeHead(200, {"Content-Type": "image/jpeg"});

if (array.length == 2 )

{

img = fs.readFileSync('pic/' + array[0] + array[1] + '.jpg');

}

else

{

var num = Math.floor(Math.random() \* 10);

img = fs.readFileSync('pic/' + data + num + '.jpg');

}

response.write(img);

response.end();

}

exports.start = start;

exports.crane = crane;

exports.toad = toad;

exports.giraffe = giraffe;

#### Server.JS

var http = require("http");

var fs = require("fs");

var url = require("url");

var query = require("querystring");

function start(route, handle) {

function onRequest(request, response) {

var postData = "";

var pathname = url.parse(request.url).pathname;

console.log("Request for " + pathname + " received.");

request.setEncoding("utf8");

request.addListener("data", function(postDataChunk) {

postData += postDataChunk;

pathname ="/"+ query.parse(postData).text.split(' ')[0];

console.log("Received POST data chunk '"+

postDataChunk + "'.");

});

request.addListener("end", function() {

route(handle, pathname, response, postData);

});

}

http.createServer(onRequest).listen(8888);

console.log("Server has started.");

}

exports.start = start;

#### Index.JS

var server = require("./server");

var router = require("./router");

var requestHandlers = require("./requestHandlers");

var handle = {}

handle["/"] = requestHandlers.start;

handle["/crane"] = requestHandlers.crane;

handle["/toad"] = requestHandlers.toad;

handle["/giraffe"] = requestHandlers.giraffe;

server.start(router.route,handle);

#### router.JS

function route(handle, pathname, response, postData) {

console.log("About to route a request for " + pathname);

if (typeof handle[pathname] === 'function') {

handle[pathname](response, postData);

} else {

console.log("No request handler found for " + pathname);

response.writeHead(404, {"Content-Type": "text/plain"});

response.write("404 Not found");

response.end();

}

}

exports.route = route;

#### requestHandlers.JS

var fs = require("fs");

var query = require('querystring');

function start(response, postData) {

console.log("Request handler 'start' was called.");

var body = '<html>'+

'<head>'+

'<meta http-equiv="Content-Type" content="text/html; '+

'charset=UTF-8" />'+

'</head>'+

'<body>'+

'<form onsubmit="return check()" method="post">'+

'<textarea name="text" rows="10" cols="40"></textarea>'+

'<input type="submit" value="Submit text" />'+

'</form>'+

'</body>'+

'<script>'+

'check();'+

'</script>'+

'</html>';

response.writeHead(200, {"Content-Type": "text/html"});

response.write(body);

response.end();

}

function crane(response, postData) {

console.log("Request handler 'crane' was called.");

upload(response, postData);

}

function toad(response, postData) {

console.log("Request handler 'toad' was called.");

upload(response, postData);

}

function giraffe(response, postData) {

console.log("Request handler 'giraffe' was called.");

upload(response, postData);

}

function check() {

var inputwords = ['crane','toad','giraffe'];

var input = document.getElementById("input");

var inputword = input.value;

var array = inputword.split(' ');

for (var i = 0; i < inputwords.length; i++) {

if (inputword == inputwords[i]) {return true;}

else {

for (var j = 0; j < 10; j++) {

if (inputword == inputwords[i] + ' ' + j) {return true;}

}

}

}

alert ("Неверный ввод данных");

return false;

}

function upload(response, postData) {

console.log("Request handler 'upload' was called.");

var data = query.parse(postData).text;

var array = data.split(' ');

var img;

response.writeHead(200, {"Content-Type": "image/jpeg"});

if (array.length == 2 )

{

img = fs.readFileSync('pic/' + array[0] + array[1] + '.jpg');

}

else

{

var num = Math.floor(Math.random() \* 10);

img = fs.readFileSync('pic/' + data + num + '.jpg');

}

response.write(img);

response.end();

}

exports.start = start;

exports.crane = crane;

exports.toad = toad;

exports.giraffe = giraffe;

### Выводы

В ходе выполнения лабораторной работы реализованы ответы обработчиков и POST-запросы.